# Improving Machine Translation Quality Prediction with Syntactic Tree Kernels

# Improving Machine Translation Quality Prediction with Syntactic Tree Kernels

**Christian Hardmeier**
Uppsala universitet
Inst. för lingvistik och filologi
SE-751 26 Uppsala
christian.hardmeier@lingfil.uu.se

## Abstract

We investigate the problem of predicting the quality of a given Machine Translation (MT) output segment as a binary classification task. In a study with four different data sets in two text genres and two language pairs, we show that the performance of a Support Vector Machine (SVM) classifier can be improved by extending the feature set with implicitly defined syntactic features in the form of tree kernels over syntactic parse trees. Moreover, we demonstrate that syntax tree kernels achieve surprisingly high performance levels even without additional features, which makes them suitable as a low-effort initial building block for an MT quality estimation system.

## 1 Introduction

Even though automatic high-quality translation for general domains is still far beyond the reach of current Statistical Machine Translation (SMT) systems, recent systems achieve levels of performance that make them viable for use as core elements in commercial translation processes. In certain text genres and with systems trained on sufficient amounts of in-domain data, producing raw translations with an SMT system and having human translators post-edit them can be a more cost-effective way of obtaining production-quality translations than doing fully human translation. One example for this is the domain of TV film subtitles, where a good SMT system can output translations closely similar to a translation produced by a professional translator for more than 30 % of the subtitles under favourable conditions (Volk et al., 2010). Still, despite the large number of good translations, other subtitles in the SMT output can be of very low quality, placing an unnecessary burden on the post-editors, who have to take a decision to discard the bad raw translation before translating the subtitle from scratch anyway.

In order to reduce the effort post-editors have to spend on acceptance decisions and make subtitle post-editing a more pleasant experience, it would be desirable to predict the quality of a segment automatically given the input, the output and the models of the SMT system, a task that has gone under the name of confidence prediction in the literature. While the SMT system itself internally scores alternative translations of each segment to find the best one, raw SMT scores are not sufficient as a confidence measure. As conditional probabilities, they are not comparable across sentences. Furthermore, they are not properly normalised by the SMT decoder since performing the required normalisation would render decoding intractable. Using a decoder-external confidence prediction module also makes it possible to use certain features which by their nature are difficult to integrate in the left-to-right beam search framework typical of current phrase-based SMT decoders.

Previous research in Machine Translation (MT) confidence estimation has used a variety of different features representing characteristics of the input and the output sentence, their relation with each other as well as the relation of the input sentence to the training data. Successful feature sets are the result of considerable engineering effort; feature extraction requires a collection of tools and models dealing with various aspects of the texts that might affect translation quality. In the present paper, we explore the use of syntactic tree kernels over parsed representations of MT input and output strings in conjunction with Support Vector Machine (SVM) classification for MT confidence estimation. Tree kernels are interesting since they allow us to define implicitly an immense space of structural features and leave the feature selection



Mikel L. Forcada, Heidi Depraetere, Vincent Vandeghinste (eds.)
*Proceedings of the 15th Conference of the European Association for Machine Translation*, p. 233–240
Leuven, Belgium, May 2011

problem to the SVM training algorithm. Structural sentence characteristics are likely to be at the root of many important problems, such as word reordering, which is notoriously difficult for SMT, but selecting the right structural features manually is difficult and tedious. Tree kernels are ideal as an initial building block of an MT confidence system as they provide reasonable performance with minimal effort – it is sufficient to parse the data to get started.

In this paper, we focus on the task of filtering out presumably bad translation from SMT output using binary SVM classifiers. For four different datasets in two language pairs and two text genres, we build and evaluate classifiers based on explicitly extracted feature sets, syntactic tree kernels and their combination. We demonstrate that it is relatively easy to build a reasonable classifier using the tree kernel approach alone and that syntactic tree kernels have something to contribute even in the presence of a traditional feature set.

## 2 Related work

The problem of sentence-level confidence estimation for Machine Translation has been addressed with various Machine Learning techniques in the past. Blatz et al. (2004) present a comparison of different Machine Learning algorithms for MT confidence estimation and a set of features that has become the basis of much later work. They train classifiers trained on data labelled automatically based on the NIST and WER Machine Translation evaluation measures, accepting as good the top-scored 5 or 30 percent of the examples. A similar setup and feature set were used by Quirk (2004), who also ran some experiments with a very small manually annotated corpus, using only 350 sentences for training and 150 sentences for testing. A comparable feature set was also used by Soricut and Echihabi (2010).

Specia et al. (2009a) use a fairly large feature set including most of the features proposed by Blatz et al. (2004) to train a Partial Least Squares (PLS) regressor on a variety of datasets, both manually and automatically annotated. In another paper from the same year (Specia et al., 2009b), they suggest a way to compute a threshold value to use the PLS regressor as classifier at a given target precision using Inductive Confidence Machines. They argue that if the MT output is to be post-edited by professional translators, it may be more important to ensure a reasonable level of precision by suppressing bad translations to avoid flooding the translators with bad MT output than to achieve high levels of recall. While this is an important point to consider, it seems at least doubtful, and very much dependent on the particularities of a given workflow, whether filtering out bad translations with a recall of less than 30 %, as reported in some of their experiments, is really making the best use of an existing MT system. The research in these papers was later published as a journal article (Specia et al., 2010b), which is interesting for us because it reports some evaluation figures directly comparable to our work.

Work presented in the papers discussed so far has used explicitly engineered features based on various aspects of the input and output but not requiring syntactic parsing. Parse tree information has been used e. g. by Liu and Gildea (2005), who use a BLEU-inspired measure of parse tree similarity as well as Subset Tree Kernels (Collins and Duffy, 2001) in the context of MT evaluation, i. e. for scoring against a reference translation. They do not train an SVM or a similar Machine Learning algorithm with their tree kernels; instead, the tree kernel function is directly used to measure the similarity between a candidate and a reference translation. For this purpose, the BLEU-inspired "subtree metric" proposed by the authors works much better than the tree kernel function.

In an MT confidence estimation task, parse tree features were used by Gamon et al. (2005). They trained an SVM classifier to predict whether a sentence was more likely produced by a human or by an MT system, under the assumption that "machine-translated output is known a priori to be of much worse quality than human translations." This assumption is questioned by Specia et al. (2009a). Parse tree information is encoded as a set of binary features indicating the presence or absence of particular context-free productions. Some semantic features are also included.

In our experiments, we adopt the experimental setup of Specia et al. (2009b) in terms of the data used and most parts of the experimental protocol. Unlike them, however, we train binary classifiers with Support Vector Machines rather than PLS regressors and strive for balanced precision and recall scores. In terms of features, the main contribution of our work is the use of tree kernels as a way to define a large implicit feature space poten-

tially covering abstract linguistic phenomena with relatively low effort compared to the explicit feature engineering approach of previous work.

## 3 Datasets

The research presented in this paper was mainly developed while working on a confidence estimation component for an MT system for film subtitles, for which we had a specific dataset freshly annotated with quality scores at our disposal. Our annotations were modelled after a collection of annotated data published by Specia et al. (2010a), on which we ran our experiments for comparison since the subtitle dataset cannot be made publicly available.

### 3.1 Europarl datasets

The data collection provided by Specia et al. (2010a) is composed of 4,000 sentences randomly drawn from the development and test sets of the WMT 2008 Machine Translation shared task, translated from English into Spanish with four different Statistical Machine Translation systems. The quality of the MT output for each single sentence was judged by professional translators on a scale ranging from 1 to 4 with the following definitions (Specia et al., 2010a):

1. requires complete retranslation
2. a lot of post-editing needed (but quicker than retranslation)
3. a little post-editing needed
4. fit for purpose

The datasets are distributed in lowercased and tokenised form.

In this paper, we report experimental results only for systems 1, 2 and 3 of this collection. System 4 is a very unbalanced set with 93.5 % of the examples belonging to the negative class. Like Specia et al. (2010b), who used the same data collection, we observed that classifiers trained on this data almost invariably learn to reject everything they see, so these results are fairly uninteresting and therefore omitted here.

### 3.2 Subtitle dataset

Our subtitle dataset was composed of the subtitle captions of 12 episodes of different TV series, which had been translated from their original language English into Swedish with a phrase-based SMT system and then post-edited by professional translators to achieve a sufficient quality level to allow broadcasting the results. The total number of subtitles (segments) amounted to 4,442, of which 1,363 (3 files) had been post-edited independently by three different persons, whose scores had been averaged, while the other 3,079 subtitles (9 files) had been post-edited by one person only. The post-editors had been asked to judge the quality of the raw MT output, assigning to each subtitle a score between 1 and 4. The definitions of the scores were very similar to those used by Specia et al. (2010a), except for the fact that the definition of grade 3 had been slightly modified to focus more clearly on post-editing speed, and the two intermediate grades were illustrated with clarifying sentences to make their use more consistent. The instructions given to the post-editors were as follows:

1. MT output unusable, subtitle needs to be retranslated from scratch.
2. Post-editing quicker than retranslation.
   ("I needed to think about whether or not the MT output was usable.")
3. Only quick post-editing required.
   ("I could see almost immediately what I had to change.")
4. MT output fit for purpose, no changes required.

Our experiments were set up as binary classifiers. Scale grades 1 and 2 were considered negative, 3 and 4 positive examples.

Unfortunately, the inter-annotator agreement achieved on the portion annotated by three post-editors was relatively low. Agreement as measured by Krippendorff's $\alpha$ (Krippendorff, 2004) for ordinally scaled data reached 0.495 for the 4-class data and 0.319 after collapsing categories. There was considerable variation between the individual subtitle files, which we suspect is due partly to the fact that the film episodes came from different genres and presented different challenges to the SMT system and partly to the circumstance that the set of annotators scoring the files varied.

## 4 Feature extraction

### 4.1 Explicit features

As a baseline system, we created a classifier based on a number of features explicitly extracted from the datasets. Our feature set was modelled on a

subset of the features used by Specia et al. (2009b). It contained the following items:

- number of words, length ratio
- type-token ratio
- number of tokens matching particular patterns:
  - numbers
  - opening and closing parentheses
  - strong punctuation signs
  - weak punctuation signs
  - ellipsis signs
  - hyphens
  - single and double quotes
  - apostrophe-s tokens
  - short alphabetic tokens ($\leq 3$ letters)
  - long alphabetic tokens ($\geq 4$ letters)
- source and target language model (LM) and log-LM scores
- LM and log-LM scores normalised by sentence length
- number and percentage of out-of-vocabulary words
- percentage of source 1-, 2-, 3- and 4-grams occurring in the source part of the training corpus
- percentage of source 1-, 2-, 3- and 4-grams in each frequency quartile of the training corpus

Whenever applicable, features were computed for both the source and the target language, and additional features were added to represent the squared difference of the source and target language feature values.

For the subtitle dataset only, we ran some experiments with an extended feature set containing a number of additional features:

- number of some particular tokens specific to subtitles
  - discourse turn marker
  - marker for continuation in next subtitle
  - marker for continuation from previous subtitle
- a binary feature indicating that the output contains more than three times as many alphabetic tokens as the input
- percentage of unaligned words and words with $1:1$, $1:n$, $n:1$ and $m:n$ alignments.

These features were not used in the Europarl experiments, partly because they were not applicable to the genre and tokenisation of those datasets, partly because alignment information from the MT decoder, which we used for computing the alignment features, was not provided in the datasets.

### 4.2 Parse trees

We annotated all datasets with both parse trees for both the source and the target language. In the source language, English, we were able to produce both constituency and dependency parses. In the target languages, Swedish and Spanish, we limited our experiments to dependency parses because of the better availability of parsing models. English constituency parses were produced with the Stanford parser (Klein and Manning, 2003) using the model bundled with the parser. For dependency parsing, we used the MaltParser (Nivre et al., 2006). POS tagging was done with HunPOS (Halácsy et al., 2007) for English and Swedish and SVMTool (Giménez and Márquez, 2004) for Spanish, with the models provided by the OPUS project (Tiedemann, 2009). A recaser based on the Moses SMT system (Koehn et al., 2007) and trained on the WMT 2008 training data was used to transform the lowercase-only Europarl datasets into mixed-case form before tagging and parsing.

The MT output was parsed with a standard parser model trained on regular treebank data. SMT output contains many grammatically malformed sentences. We do not know of a reliable method to assess the impact of this problem on parsing accuracy, nor is it clear what effect reduced parsing accuracy has on classifier performance, since the tree-kernel classifier may very well be able to extract useful information from corrupted parse trees if the corruption is sufficiently systematic. In the present work, we therefore treat the parsers as a black box and rely on the classifier to make sense of whatever input it receives.

To be used with tree kernels, the output of the dependency parser had to be transformed into a single tree structure with a unique label per node and unlabelled edges, similar to a constituency parse tree. We followed Johansson and Moschitti (2010) in using a tree representation which encodes part-of-speech tags, dependency relations and words as sequences of child nodes (see fig. 1).

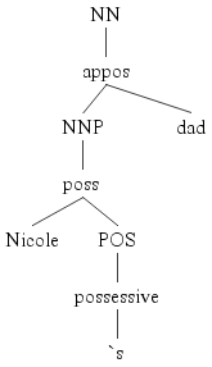
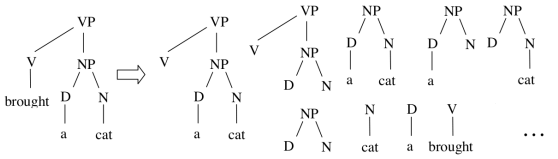
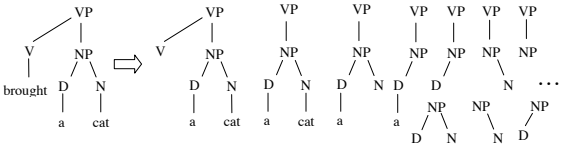



Figure 1: Representation of the dependency tree fragment for the words *Nicole 's dad*




A tree and some of its Subset Tree Fragments




A tree and some of its Partial Tree Fragments

Figure 2: Tree fragments extracted by the Subset Tree Kernel and by the Partial Tree Kernel. Illustrations by Moschitti (2006a).

## 5 Implicit feature modelling with tree kernels

To exploit parse tree information in our Machine Learning (ML) component, we used tree kernel functions. Kernel functions make it possible to represent very complex and high-dimensional feature spaces for certain ML techniques such as Support Vector Machines (SVM) in an efficient way. They take advantage of the fact that in the learning and inference algorithms for these ML methods, feature vectors are only ever evaluated in the form of dot products over pairs of data points. Dot products over certain types of feature spaces can be computed very efficiently without reference to the full feature representation.

Tree kernels (Collins and Duffy, 2001) are kernel functions defined over pairs of tree structures. They measure the similarity between two trees by counting the number of common substructures. Implicitly, they define an infinite-dimensional feature space whose dimensions correspond to all possible tree fragments. Features are thus available to cover different kinds of abstract node configurations that can occur in a tree. The important feature dimensions are effectively selected by the SVM training algorithm through the selection and weighting of the support vectors.

In our experiments, we used two different kinds of tree kernels (see fig. 2). The Subset Tree Kernel (Collins and Duffy, 2001) considers tree fragments consisting of more than one node with the restriction that if one child of a node is included, then all its siblings must be included as well so that the underlying production rule is completely represented. This kind of kernel is well suited for constituency parse trees and was used in our experiments with constituency trees. For the experiments with dependency trees, we used the Partial Tree Kernel (Moschitti, 2006a) instead. It extends the Subset Tree Kernel by permitting also the extraction of tree fragments comprising only part of the children of any given node. Lifting this restriction makes sense for dependency trees since a node and its children do not correspond to a grammatical production in a dependency tree in the same way as they do in a constituency tree.

## 6 Experiments and results

All our experiments were run with the SVMlight software with tree kernel extensions (Moschitti, 2006b; Joachims, 1999), using polynomial kernels of degree 3 for the explicit features. In experiments with both a polynomial kernel and a tree kernel, a linear combination with equal weights was used. Each of the results obtained was obtained by randomly subsampling the complete dataset five times, dividing it into a training part (80 %) and a test part (20 %). The figures reported are the means of precision, recall and F1 score over the five runs for a binary classifier separating positive examples labelled 3 or 4 by the annotators from negative examples labelled 1 or 2.

The experimental results are presented in tables 1 (Europarl datasets) and 2 (subtitle dataset). Baseline scores were calculated for a majority class classifier which simply labels all examples as positive. This results in a precision equal to the propor-

| | System 1 | | | System 2 | | | System 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F | P | R | F | P | R | F |
| majority class | 71.0 | 100.0 | 83.0 | 54.6 | 100.0 | 70.6 | 51.8 | 100.0 | 68.3 |
| explicit features | 73.2 | 96.7 | 83.5 | 67.1 | 82.7 | 74.0 | 74.5 | 66.7 | 70.4 |
| explicit + constituency | 80.2 | 90.7 | **85.1** | 74.4 | 73.3 | 73.9 | 73.6 | 73.2 | 73.4 |
| explicit + dependency (src/tgt) | 78.0 | 92.9 | 84.8 | 74.1 | 76.2 | **75.1** | 73.8 | 74.0 | **73.9** |

Table 1: Experimental results (Precision/Recall/F-score) for the Europarl datasets

| | P | R | F |
|---|---|---|---|
| majority class | 50.2 | 100.0 | 66.8 |
| all features | 69.5 | 58.3 | 63.3 |
| reduced features | 72.3 | 48.1 | 57.7 |
| all + constituency (S) | 67.5 | 66.3 | 66.8 |
| all + dependency (S+T) | 68.7 | 68.8 | **68.8** |
| red. + constituency (S) | 68.2 | 67.8 | 68.0 |
| red. + dependency (S+T) | 68.3 | 67.6 | 67.9 |

Table 2: Experimental results (Precision/Recall/F-score) for the subtitle dataset

tion of positive examples in the dataset and, trivially, in a recall score of 100 %. It turns out that this baseline is relatively hard to beat in terms of balanced F-score for some datasets. This does not necessarily mean that a classifier with a lower performance is useless. Depending on the application scenario, it may be more important to obtain higher precision at the cost of somewhat lower recall in order to make the post-editors' job less tedious. This is the stance adopted by Specia et al. (2009b), who argue that more experienced translators make high demands on the quality of MT output, so "a larger proportion of positive examples" must potentially be discarded.

For the Europarl systems, classifiers based on the reduced explicit feature set we applied to all systems performed slightly better than the baseline, with gains ranging from 0.5 points in F-score for system 1 to 3.4 points for system 2. For the subtitle dataset, this is not the case: The performance of the reduced feature set, which is identical to the feature set used by the classifiers in the Europarl experiments, is more than 9 points below the baseline. By including the additional features listed at the end of section 4.1, the F-score can be improved from 57.7 % to 63.3 %, but it remains several points below the baseline of 66.8 %.

Several factors may have contributed to the low performance of the explicit feature set on the subtitle data. To begin with, the sentences in the subtitle data set are much shorter than the sentences in the Europarl datasets and mostly written in a casual oral style characterised, among other things, by low syntactic complexity. This may have the effect that some of our features that are supposed to measure sentence complexity in a crude way, such as the counts of various punctuation tokens, have little of interest to measure. The vocabulary coverage of the subtitle translation system is generally quite good and out-of-vocabulary words, when they occur, are often proper names that can be translated correctly by just copying them to the output, so the vocabulary coverage features may be less useful than in texts where out-of-vocabulary items are more frequent. Finally, the subtitle MT system is known to suffer from a specific problem that causes it to drop content words occasionally. Probably some of our additional features help detect items affected by this particular bug, partly explaining the difference in performance between the full and the reduced feature set.

The best overall performance is obtained by combining the explicit feature set with tree kernels (tables 1 and 2). All experiments in these configurations performed at least as well, and almost always better, than either the trivial baseline or the classifiers with explicit features only. It is not clear whether the constituency or the dependency parse configuration is to be preferred, but the former has the advantage that it reaches similar levels of performance without parsing the MT output at all.

In table 3, we show the results of all experiments in terms of accuracy. While we believe that precision and recall scores are more informative, this format has the advantage of being comparable with the scores published by Specia et al. (2010b) for the three Europarl test sets. As can be seen, our systems are generally competitive with the results published in the recent literature. This table also contains results for a number of systems that use only tree kernels and do not make use of the explicit features at all. For these experi-

| | Europarl | | | sub- |
|---|---|---|---|---|
| | 1 | 2 | 3 | titles |
| majority class | 71.0 | 54.6 | 51.8 | 50.2 |
| Specia et al. (2010b), best results | 76.8 | 66.0 | 69.8 | |
| explicit features, full set | | | | 66.4 |
| explicit features, reduced set | 72.6 | 68.7 | 70.3 | 64.3 |
| constituency tree kernel (src) | | 66.4 | 66.9 | 64.7 |
| dependency tree kernel (src) | | 67.6 | 66.6 | 64.0 |
| dependency tree kernel (tgt) | | 65.5 | 65.2 | 62.6 |
| dependency tree kernel (src+tgt) | | 66.4 | 67.8 | 65.0 |
| full set + constituency (src) | | | | 66.7 |
| full set + dependency (src+tgt) | | | | **68.3** |
| reduced set + constituency (src) | **77.8** | 71.1 | 72.5 | 65.6 |
| reduced set + dependency (src+tgt) | 76.7 | **72.4** | **72.8** | 67.9 |

Table 3: Experimental results in terms of accuracy

ments, the scores of Europarl system 1 are omitted because the tree-kernel-only classifiers degenerated into the uninteresting accept-all case for this dataset, and small score differences with respect to the majority class baseline are exclusively due to the sampling variance.

While the results of the tree-kernel-only systems were generally lower than the corresponding results obtained with the explicit feature set, it is interesting to notice that this was the case only by a relatively small margin. The constituency parse configuration performs well even though it only uses information from the source language. For the dependency parses, using only the source language works slightly better than using only the target language, and combining the two generally works best. Taking into account the fact that setting up the tree-kernel-only systems only requires a working parser for one or both languages, whereas constructing explicit feature sets takes a considerable amount of engineering work, it seems reasonable to use tree kernels as an initial building block for a new MT confidence estimation system that can deliver a certain level of performance on its own, adding other features as required to improve performance.

## 7 Conclusions

Syntactic tree kernels are an easy way to exploit complex structural information in a Machine Learning system. This is especially true when using a constituency parser whose output can directly be fed into the ML component, but dependency trees can also be used after a simple conversion step. The feature space expressed by syntax trees is very expressive, and feature selection can be handled effectively by the SVM training algorithm. In combination, these advantages make a tree-kernel-based approach a perfect starting point for an MT quality prediction system. This is borne out by our experimental results, which show that MT quality classifiers based on tree kernels alone perform only slightly worse than traditional systems based on explicit features while being considerably easier to build.

This is not to say, of course, that explicit features have nothing to contribute. Our best results were obtained by combining syntactic tree kernels with a traditional feature set, and this is not surprising considering that the tree kernels we used only encode information about the MT input and output segments in isolation and do not take into account their relation to the SMT training data or their mutual relation with each other. At least the latter point could certainly be addressed with a more advanced tree kernel design as well, and it remains for future work to show whether this may lead to further improvements. For the time being, it is safe to conclude that tree kernels should have their place in MT quality estimation as an easy and versatile method to encode complex feature sets.

## Acknowledgements

Parts of this work were carried out while the author was working at Fondazione Bruno Kessler, HLT unit, Trento, Italy. We gratefully acknowledge the help of Alessandro Moschitti, who gave advice on using tree kernel classifiers, of Jörgen

Aasa, who organised the preparation of our subtitle dataset, and of Lucia Specia and Marco Turchi, who explained many details of their own experimental setup.

## References

Blatz, John, Erin Fitzgerald, George Foster, et al. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, pages 315–321.

Collins, Michael and Nigel Duffy. 2001. Convolution kernels for natural language. In *Proceedings of NIPS 2001*, pages 625–632.

Gamon, Michael, Anthony Aue, and Martine Smets. 2005. Sentence-level MT evaluation without reference translations: Beyond language modeling. In *Proceedings of the 10th Annual Conference of the EAMT*, pages 103–111, Budapest.

Giménez, Jesús and Lluís Márquez. 2004. SVMTool: A general POS tagger generator based on Support Vector Machines. In *Proceedings of the 4th Conference on International Language Resources and Evaluation (LREC-2004)*, Lisbon.

Halácsy, Péter, András Kornai, and Csaba Oravecz. 2007. HunPos – an open source trigram tagger. In *Proceedings of the 45th Meeting of the Association for Computational Linguistics*, pages 209–212, Prague.

Joachims, Thorsten. 1999. Making large-scale SVM learning practical. In Schölkopf, B., C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*. MIT Press.

Johansson, Richard and Alessandro Moschitti. 2010. Syntactic and semantic structure for opinion expression detection. In *Proceedings of the 2010 Conference on Natural Language Learning*, Uppsala.

Klein, Dan and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, et al. 2007. Moses: open source toolkit for statistical machine translation. In *Annual meeting of the Association for Computational Linguistics: Demonstration session*, pages 177–180, Prague.

Krippendorff, Klaus. 2004. Measuring the reliability of qualitative text analysis data. *Quality and Quantity*, 38:787–800.

Liu, Ding and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation*, pages 25–32, Ann Arbor.

Moschitti, Alessandro. 2006a. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of the 17th European Conference on Machine Learning*, Berlin.

Moschitti, Alessandro. 2006b. Making tree kernels practical for natural language learning. In *Proceedings of the Eleventh International Conference of the European Association for Computational Linguistics*, Trento.

Nivre, Joakim, Johan Hall, and Jens Nilsson. 2006. MaltParser: A language-independent system for data-driven dependency parsing. In *Proceedings of the 5th Conference on International Language Resources and Evaluation (LREC-2006)*, pages 2216–2219, Genoa.

Quirk, Christopher B. 2004. Training a sentence-level machine translation confidence measure. In *Proceedings of the 4th Conference on International Language Resources and Evaluation (LREC-2004)*, pages 825–828, Lisbon.

Soricut, Radu and Abdessamat Echihabi. 2010. TrustRank: Inducing trust in automatic translations via ranking. In *Proceedings of the 48th Meeting of the Association for Computational Linguistics*, pages 612–621, Uppsala.

Specia, Lucia, Nicola Cancedda, Marc Dymetman, Marco Turchi, and Nello Cristianini. 2009a. Estimating the sentence-level quality of Machine Translation systems. In *Proceedings of the 13th Annual Conference of the EAMT*, pages 28–35, Barcelona.

Specia, Lucia, Craig Saunders, Marco Turchi, Zhuoran Wang, and John Shawe-Taylor. 2009b. Improving the confidence of Machine Translation quality estimates. In *Proceedings of MT Summit XII*, Ottawa.

Specia, Lucia, Nicola Cancedda, and Marc Dymetman. 2010a. A dataset for assessing machine translation evaluation metrics. In *Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC-2010)*, pages 3375–3378, Valletta, Malta.

Specia, Lucia, Dhwaj Raj, and Marco Turchi. 2010b. Machine translation evaluation versus quality estimation. *Machine Translation*, 24:39–50.

Tiedemann, Jörg. 2009. News from OPUS – a collection of multilingual parallel corpora with tools and interface. In Nicolov, N., K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, pages 237–248. John Benjamins, Amsterdam.

Volk, Martin, Rico Sennrich, Christian Hardmeier, and Frida Tidström. 2010. Machine translation of TV subtitles for large scale production. In *Proceedings of the Second Joint EM+/CNGL Workshop "Bringing MT to the User: Research on Integrating MT in the Translation Industry" (JEC 2010)*, pages 53–62, Denver, CO.